JAVA.MATH.ROUNDINGMODE ENUMERATION

Introduction

The **java.math.RoundingMode** enumeration specifies a rounding behavior for numerical operations capable of discarding precision. Each rounding mode indicates how the least significant returned digit of a rounded result is to be calculated.

If fewer digits are returned than the digits needed to represent the exact numerical result, the discarded digits will be referred to as the discarded fraction regardless the digits' contribution to the value of the number. In other words, considered as a numerical value, the discarded fraction could have an absolute value greater than one.

This enum is intended to replace the integer-based enumeration of rounding mode constants in BigDecimal (BigDecimal.ROUND_UP, BigDecimal.ROUND_DOWN, etc.).

Enum declaration

Following is the declaration for **java.math.RoundingMode** enum:

```
public enum RoundingMode
    extends Enum<RoundingMode>
```

Constants

Following are the constants for **java.math.RoundingMode** enum:

- **CEILING** -- Rounding mode to round towards positive infinity.

- **DOWN** -- Rounding mode to round towards zero.

- **FLOOR** -- Rounding mode to round towards negative infinity.

- **HALF_DOWN** -- Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round down.

- **HALF_EVEN** -- Rounding mode to round towards the "nearest neighbor" unless both neighbors are equidistant, in which case, round towards the even neighbor.

- **HALF_UP** -- Rounding mode to round towards "nearest neighbor" unless both neighbors are equidistant, in which case round up.

- **UNNECESSARY** -- Rounding mode to assert that the requested operation has an exact result, hence no rounding is necessary.

- **UP** -- Rounding mode to round away from zero.

Enum methods

S.N.	Method & Description
1	**static RoundingMode valueOf(int rm)**

	This method returns the RoundingMode object corresponding to a legacy integer rounding mode constant in BigDecimal.
2	**static RoundingMode valueOf(String name)** This method returns the enum constant of this type with the specified name.
3	**static RoundingMode[ ] values()** This method returns an array containing the constants of this enum type, in the order they are declared.

Example

The following example shows the usage of math.RoundingMode methods

```java
package com.tutorialspoint;

import java.math.*;

public class RoundingModeDemo {

public static void main(String[] args) {

 // create 2 RoundingMode objects
 RoundingMode rm1, rm2;

 // create and assign values to rm and name
 int rm = 5;
 String name = "UP";

 // static methods are called using enum name

 // assign the the enum constant of rm to rm1
 rm1 = RoundingMode.valueOf(rm);

 // assign the the enum constant of name to rm2
 rm2 = RoundingMode.valueOf(name);

 String str1 = "Enum constant for integer " + rm + " is " +rm1;
 String str2 = "Enum constant for string " + name + " is " +rm2;

 // print rm1, rm2  values
 System.out.println( str1 );
 System.out.println( str2 );

 String str3 = "Enum constants of RoundingMode in order are :";

 System.out.println( str3 );

 // print the array of enum constatnts using for loop
 for (RoundingMode c : RoundingMode.values())
 System.out.println(c);
    }
}
```

Let us compile and run the above program, this will produce the following result:

```
Enum constant for integer 5 is HALF_DOWN
Enum constant for string UP is UP
Enum constants of RoundingMode in order are :
UP
DOWN
CEILING
FLOOR
HALF_UP
HALF_DOWN
HALF_EVEN
UNNECESSARY
```